

OPTIMASI PENGAMBILAN DAN PENATAAN ULANG BARANG DI GUDANG DENGAN PENERAPAN STACK MENGGUNAKAN METODE *GENETIC ALGORITHM*

Resmana Lim, Kartika Gunadi

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
e-mail : resmana@peter.petra.ac.id, kgunadi@peter.petra.ac.id

Ong Wie Gang

Alumni Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
e-mail : wie_gang@yahoo.com

ABSTRAK: Dunia industri yang semakin maju membutuhkan komputer sebagai alat bantu utama. Sistem komputerisasi di dalam dunia industri akan membuat beberapa hal menjadi lebih mudah, antara lain dalam pengambilan barang dan penataan ulang barang di gudang. Suatu perangkat lunak untuk keperluan tersebut telah dibuat oleh penulis. Perangkat lunak yang dibuat ini dikhususkan untuk gudang yang barang-barangnya disimpan dalam palet-palet pada rak di gudang. Penataan ulang barang ini menggunakan algoritma genetika, yang akan melalui proses pembuatan populasi awal, pencarian *fitness cost* dengan memperhitungkan luas area dan berat barang, pengurutan data, imigrasi, persilangan, dan mutasi. Melalui perangkat lunak ini, *user* juga dapat melihat kondisi gudang secara baik secara 2 dimensi ataupun 3 dimensi, yang dalam pengerjaannya menggunakan OpenGL. Sistem dari perangkat lunak yang dibuat ini sudah diuji coba dan memberikan nilai rata-rata *fitness cost* di atas 90%, ini berarti perangkat lunak ini telah sukses dalam menjalankan fungsinya.

Kata kunci: penataan barang, pengambilan barang dari gudang, penataan ulang barang, rak, palet, Algoritma Genetika, imigrasi, persilangan, mutasi, OpenGL.

ABSTRACT: Industrial field which getting more advance needs computer to help them. In industrial field, computer will make something get easier, fetching and reordering goods in warehouse is including here. A software including that needs have been made by writers. This software is specified for a warehouse which goods stored in plates at the rack in the warehouse. To reordering goods, writers use the Genetic Algorithm, which will process making population, finding the fitness cost with counting the vast and weight of the goods, sorting, immigration, cross over, and mutation. With this software, user can also check the condition of the warehouse in two and three dimensions, using OpenGL. This optimization software have been tried and give fitness cost average up to 90%, this software have been succeed to operate the function.

Keywords: ordering goods, to fetch goods, reordering goods, plate, rack, Genetic Algorithm, immigration, cross over, mutation, OpenGL.

1. PENDAHULUAN

Pada saat ini komputer merupakan kebutuhan yang umum dalam sebuah perusahaan. Di dalam perusahaan, banyak hal menjadi lebih efisien dengan menggunakan sistem komputerisasi. Salah satu bentuk perusahaan yang sangat terbantu dengan

adanya komputer adalah perusahaan dalam bidang industri.

Pada perusahaan industri khususnya yang memproduksi suatu barang, selalu terjadi proses produksi, penyimpanan, dan penjualan hasil produksi. Lebih jauh lagi, pada tulisan ini penulis akan lebih memfokuskan pada permasalahan pengambilan barang dari dalam gudang, yang

merupakan kelanjutan dari penelitian sebelumnya milik Erwin Kristianto J., dengan judul Optimasi Pengisian Gudang dengan Penerapan Stack menggunakan Metode *Genetic Algorithm*.

Untuk mengambil suatu barang ada beberapa perhitungan yang penting untuk dilakukan, salah satunya pengaturan kembali barang-barang yang lain sehingga posisi barang-barang itu tetap optimal.

2. MODEL OPTIMASI PENATAAN, PENGAMBILAN DAN PENATAAN ULANG BARANG

2.1 Penataan Barang dalam Gudang

Di dalam dunia industri, penyimpanan barang dalam gudang memiliki berbagai macam cara, salah satu cara yang akan dibahas lebih jauh adalah dengan sistem *stack*. Pada sistem ini barang disimpan dalam suatu palet yang diletakkan pada rak. Peletakkan palet ke rak biasanya menggunakan alat bantu kendaraan *forklift*. Penelitian ini memiliki batasan:

- Gudang : Panjang : 10 – 100 meter
 Lebar : 5 – 30 meter
 Tinggi : 4 – 10 meter
 Rak : Jumlah : maksimal 20 rak
 Ukuran : 1x2 m²
 Tinggi : maks. 3 m
 Palet : Jumlah : maks. 6 palet per rak
 Tinggi : 50cm dan kelipatannya
 Beban : maks 1000 kg
 Barang : Ukuran : kelipatan 10 cm
 Tinggi : maks 290 cm

2.2 Pengambilan Barang dari Gudang

Prioritas pengambilan barang dari gudang adalah barang yang berada di ujung-ujung dari palet dan barang yang berada di pinggir palet. Setelah suatu barang diambil, perlu dilakukan penataan ulang barang-barang lain yang tidak diambil. Penataan ulang barang-barang di sini memiliki beberapa pilihan yang dapat dilakukan, antara lain penataan ulang seluruh isi gudang, penataan ulang barang-barang pada palet dimana barang yang

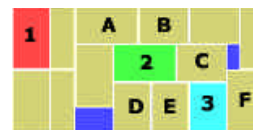
diambil tersebut berada, penataan ulang barang-barang di sekitar barang yang diambil tersebut, dan penataan ulang barang-barang yang harus dikeluarkan dahulu untuk mengambil barang tersebut.

2.3 Penataan Ulang Barang

Bila suatu barang diambil dari suatu palet, maka akan muncul beberapa kemungkinan untuk penataan ulang barang-barang yang tidak diambil.

2.3.1 Tidak Terjadi Penataan Ulang

Kemungkinan yang pertama ini bisa terjadi apabila barang yang akan diambil letaknya di ujung dari palet atau di pinggir palet. Tetapi tidak menutup kemungkinan bila dikehendaki *user*, barang yang tidak diambil untuk ditata ulang. Salah satu contoh seperti pada Gambar 1, barang berkode 1 atau 3, bila diambil tentu saja tidak perlu terjadi penataan ulang barang.



Gambar 1. Contoh Pengeluaran Barang

2.3.2 Penataan Ulang Barang yang Dikeluarkan

Untuk barang yang tidak berada di pinggir palet, bila akan diambil, maka perlu mengeluarkan dahulu barang-barang di depannya untuk menjadi jalan keluar bagi barang tersebut. Penataan ulang akan dilakukan terhadap barang-barang yang telah dikeluarkan tersebut. Dalam Gambar 1, barang 2 akan dikeluarkan. Ada dua pilihan jalan untuk mengeluarkan barang tersebut yaitu dengan mengeluarkan terlebih dahulu barang A dan B, atau barang D dan E. Tentu saja pilihan manapun yang akan diambil, barang yang dikeluarkan harus ditata kembali ke dalam palet tersebut. Penataan ulang ini bisa juga terjadi hanya ditata kembali seperti keadaan semua saja atau tidak ditata ulang.

2.3.3 Penataan Ulang Barang pada Palet

Pilihan lain *user* dalam penataan ulang barang yaitu penataan ulang semua barang yang ada pada palet yang sama dengan palet tempat barang diambil. Cara ini efektif apabila barang yang diambil dari suatu pelat jumlahnya banyak sekali.

2.3.4 Penataan Ulang Seluruh Barang di Gudang

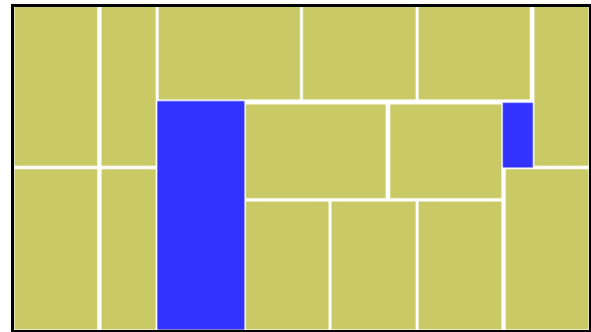
Bila dikehendaki, dapat juga seluruh barang dalam gudang untuk ditata ulang. Ini mungkin terjadi apabila setelah terjadi berkali-kali pengambilan barang tetapi tidak ditata ulang, pada saat akan mengecek stok barang maka perlu lebih dahulu untuk menata ulang barang.

2.4 Penataan Barang Baru

Bila ada barang lain yang akan dimasukkan ke dalam gudang, ada beberapa hal yang dapat dilakukan antara lain ditata pada palet-palet yang dapat ditempati barang tersebut tanpa menata ulang, ditata pada palet yang maksimal berisi persentase yang akan dimasukkan *user*, dan menata ulang seluruh palet yang memiliki tinggi yang sama dengan tinggi barang yang akan dimasukkan.

2.4.1 Tanpa Menata Ulang

Bila ada barang lain yang akan masuk ke dalam gudang maka pertama kali yang akan dilakukan adalah mencari palet yang kosong, dimana barang tersebut cukup untuk diletakkan pada palet tersebut. Seperti yang terlihat pada Gambar 2, warna biru mewakili tempat kosong. Bila barang yang akan masuk dapat ditata pada palet ini maka akan langsung dimasukkan.



Gambar 2. Contoh Palet yang Masih Dapat Diisi

2.4.2 Menata Ulang Palet

Selain menata pada palet yang masih bisa dimuati oleh barang yang akan masuk, dapat juga dilakukan penataan ulang palet. Dalam menata ulang palet ini *user* dapat memilih palet yang akan ditata ulang, bisa seluruh palet yang memiliki tinggi sama dengan tinggi barang yang akan masuk atau hanya palet yang berisikan beberapa persen barang dan memiliki tinggi sama dengan tinggi barang yang akan masuk. Persentase di sini akan diinputkan oleh *user*. Perhitungan persentase di sini adalah:

$$\text{Persentase Luas} = \frac{\text{Luas Barang}}{\text{Luas Floor}} * 100\%$$

Pada Gambar, luas barang yang dimaksud adalah yang berwarna kuning keemasan dan luas *floor* selalu tetap yaitu 2 m².

2.5 Algoritma Genetika

Inti dari algoritma genetika adalah secara bertahap akan mencari solusi terbaik (*survival of the fittest*) dari begitu banyak solusi yang ada. Pertama-tama algoritma genetika bekerja dengan membuat beberapa solusi secara acak, tentu saja dari tahapan pertama ini solusinya kemungkinan masih buruk. Solusi tersebut akan mengalami proses evolusi secara terus menerus, dan akan menghasilkan suatu solusi yang lebih baik. Setiap solusi yang terbentuk mewakili satu kromosom dan satu individu terdiri dari satu kromosom. Kumpulan dari individu-individu ini akan membentuk suatu populasi, dari populasi ini akan lahir populasi-

populasi baru sampai dengan sejumlah generasi yang ditentukan.

Gen dalam kasus ini adalah urutan barang dan posisi penempatan barang tersebut, misalkan 01H, berarti urutan barang yang pertama dan posisi penempatannya horisontal. Kromosom merupakan kumpulan dari gen-gen yang ada, yang berarti panjang kromosom akan sesuai dengan jumlah barang untuk suatu kategori barang. Sedangkan individu merupakan kumpulan kromosom, dalam kasus ini satu individu memiliki satu kromosom. Sedangkan populasi, merupakan kumpulan individu yang telah ditentukan jumlahnya oleh *user*.

Dalam eksekusinya, GA akan dilakukan untuk tiap kelompok atau kategori barang. Karena tiap barang sudah memiliki kelompok tinggi masing-masing, maka untuk tinggi barang yang bisa disimpan oleh suatu palet juga perlu pengelompokan. Pengelompokan ini dilakukan per 50 cm tinggi palet. Untuk lebih jelasnya dapat dilihat pada Tabel 1.

Tabel 1. Klasifikasi Palet dan Barang

No.	Tinggi Pelat	Tinggi Barang
1	50 cm	10 - 40 cm
2	100 cm	50 - 90 cm
3	150 cm	100 - 140 cm
4	200 cm	150 - 190 cm
5	250 cm	200 - 240 cm
6	300 cm	250 - 290 cm

Sebelum memasuki tahapan-tahapan GA terlebih dahulu akan diawali dengan perencanaan variabel-variabel yang dibutuhkan

2.5.1 Variabel-variabel GA

Variabel yang dibutuhkan dalam GA ini adalah:

- Jumlah Individu per Populasi (JI)
Semakin besar JI, semakin besar variasi individu yang dihasilkan, maka semakin besar pula kesempatan untuk mendapatkan solusi terbaik dan semakin sedikit jumlah generasi yang diperlukan untuk mendapatkan solusi terbaik.
- Jumlah Generasi (JG)

Variabel ini menentukan sampai berapa kali populasi awal akan berubah, jadi juga memiliki peran penting dalam menampilkan jumlah variasi individu, yang akan berpengaruh terhadap hasil GA.

- Tingkat Imigrasi (TI)
TI merupakan angka persentase yang akan mempengaruhi seberapa sering terjadinya imigrasi, atau munculnya individu baru dalam suatu populasi.
- Tingkat Mutasi (TM)
Variabel yang berupa angka persentase ini akan mempengaruhi seberapa banyak terjadinya mutasi dalam suatu populasi. Variabel TM merupakan salah satu variabel yang berbentuk peluang, artinya kemungkinan terjadinya mutasi dilihat tiap individunya.
- Tingkat Persilangan (*cross over*) (TP)
TP adalah peluang untuk terjadi persilangan antara sepasang individu. Dalam kenyataannya persilangan akan selalu terjadi, hanya saja jumlah gen dan gen-gen yang disilangkan akan berbeda-beda.
- Persentase Persilangan (PP)
PP merupakan angka persentase yang akan mempengaruhi berapa gen yang akan disilangkan.

2.5.2 Pembuatan Populasi Awal

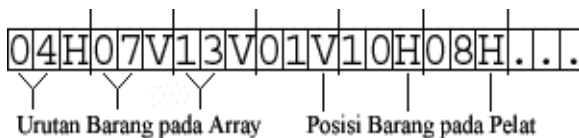
Populasi merupakan kumpulan beberapa individu. Semua populasi dalam GA ini asalnya dari 1 populasi yaitu populasi awal. Solusi atau kromosom terbaik dari populasi awal ini akan dipertahankan, dan akan mengalami proses evolusi untuk mendapatkan kemungkinan solusi yang lebih baik.

Pembuatan populasi awal ini dilakukan melalui proses pemilihan secara acak dari seluruh solusi yang ada. Pemilihan acak ini menyebabkan populasi awal dari GA tidak akan sama dalam setiap kali percobaan, meskipun semua nilai variabel yang digunakan sama.

Pada kasus pengisian barang ke gudang yang telah dibuat sebelumnya, yang menjadi individu adalah urutan barang yang akan dimasukkan ke dalam gudang. Selain urutan

barang, di dalam setiap individu juga tersimpan informasi bagaimana posisi barang ketika masuk ke dalam gudang (vertikal atau horisontal). Satu solusi akan berisi urutan dari seluruh barang yang ada. Karena itu nantinya sebelum menjalankan GA pada perangkat lunak ini, harus terlebih dahulu mengisi barang-barang yang ada beserta dengan spesifikasinya (panjang, lebar, tinggi, jumlah).

Dalam hal ini populasi adalah sebuah *string* yang berisi gen yang berjumlah sesuai dengan jumlah barang pada tiap kategori. Karena jumlah panjang gen 3 karakter, maka panjang *string* adalah $3 \times n$, dengan n merupakan jumlah barang yang akan dimasukkan ke dalam rak untuk kategori tertentu.



Gambar 3. Visualisasi Individu dalam Populasi

Semua barang yang akan disimpan pada palet-palet, terlebih dahulu akan disimpan pada sebuah array. Array tersebut berisikan informasi urutan barang tersebut pada array, dan posisinya ketika akan dimasukkan ke dalam palet.

2.5.3 Mencari Fitness Cost

Untuk mengetahui baik tidaknya solusi yang ada pada suatu individu, setiap individu pada populasi harus memiliki nilai pembandingnya (*Fitness Cost*). Melalui nilai pembanding inilah akan didapatkan solusi terbaik dengan cara pengurutan nilai pembanding dari individu-individu dalam populasi. Solusi terbaik ini akan dipertahankan, sementara solusi lain diubah-ubah untuk mendapatkan solusi yang lain, melalui tahap *cross over* dan mutasi (*mutation*).

Sebelum melakukan penempatan barang dilakukan 2 buah pengecekan terlebih dahulu, yaitu pencarian posisi barang dengan pengecekan luas palet yang masih

kosong, apakah masih bisa ditempati barang tersebut, dan pengecekan beban maksimal yang dapat ditampung oleh palet. Barang dapat masuk apabila, tempat masih kosong dan beban palet belum mencapai angka maksimal.

Pencarian posisi barang pada palet dilakukan mulai dari ujung kiri atas dari palet, dengan prioritas ke kanan, bila barang sudah tidak mungkin diletakkan di palet, maka diulangi lagi dari kiri dan diturunkan ke bawah. Pada Gambar 4 dapat dilihat lebih jelas mengenai prioritas pemilihan posisi barang pada palet.

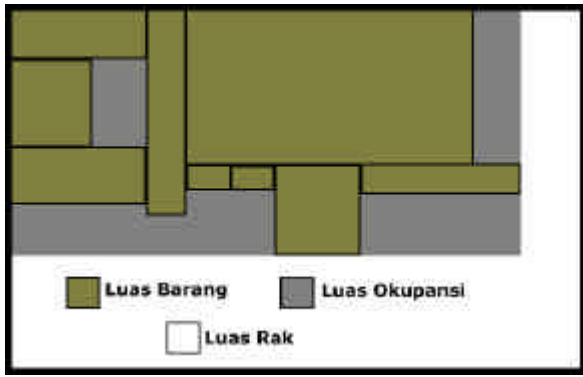
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	...	
21	22	23	24	25	...				
31	32	...							

Gambar 4. Prioritas Pemilihan Posisi Barang pada Palet

Palet horisontal maupun vertikal, prioritasnya dibuat sama, yaitu dari kiri atas ke kanan dan ke bawah. Hal ini sengaja dilakukan agar dapat dibandingkan mana yang lebih baik atau lebih sesuai dengan masalah yang dihadapi, prioritas berdasarkan panjang (sisi panjang dari palet – 2 m) atau prioritas berdasarkan lebar (sisi yang lebih pendek dari palet – 1 m).

Luas segiempat luar yang dibuat dari luas-luas barang yang ada pada sebuah palet, disebut luas okupansi, luasan ini adalah area pada palet yang dianggap terpakai dalam penyimpanan barang. Selain luas area ini, juga dicari jumlah luas barang yang telah diletakkan pada palet. Kedua luas ini akan dibandingkan, dan hasil perbandingannya dipakai sebagai *Fitness Cost*. Dalam notasi matematika dapat dituliskan sebagai berikut:

$$\text{FITNESS COST} = \frac{\text{Luas Barang}}{\text{Luas Okupansi}} * 100\%$$



Gambar 5. Kondisi Palet untuk Mencari *Fitness Cost*

2.5.4 Pengurutan Data (*Sorting*)

Tahap selanjutnya adalah tahapan pengurutan individu dalam populasi nilai pembandingan (*Fitness Cost*) yang telah didapat dari bagian sebelumnya. Tujuan utama dari pengurutan kromosom ini adalah untuk mencari individu terbaik pada suatu populasi, yang bisa dikatakan sebagai solusi terbaik sementara. Solusi yang terbaik dari kumpulan solusi terbaik sementara yang ada merupakan solusi akhir dari proses GA. Proses pengurutan kromosom ini menggunakan algoritma *quick sort*.

2.5.5 Imigrasi

Yang dimaksud imigrasi di sini adalah perginya satu individu, dan datangnya individu baru. Jadi secara acak akan dipilih satu individu misalkan:

Lama : 01H02H04H03H06V05H

Baru : 02V05V03V06H04V01V

Jadi individu lama akan dihilangkan dan individu yang baru akan digenerate ulang, untuk dimasukkan dalam populasi.

2.5.6 Tahap Cross Over

Tahapan ini akan menyilangkan dua individu yang ada dalam suatu populasi, untuk mendapatkan dua individu baru. Setelah tahap, maka akan didapat populasi baru yang jumlahnya 2 kali lipat dari populasi lama.

Individu 1 = 01H02H05V04H 03V

Individu 2 = 02H04H01V03V05V

Dari populasi yang ada, diambil individu sepasang demi sepasang untuk disilangkan (*cross over*). Persilangan pada kasus ini dilakukan dengan memindahkan sebagian urutan pada satu individu dan menukarkannya dengan individu yang lain. Ada 2 macam cara yaitu dengan *Two Points* dan *Uniform*. Pada *Two Points Cross Over*, dipilih secara acak 2 titik yang akan disilangkan.

Individu 1 = 01H | 04H01V | 04H03V

Individu 2 = 02H | 02H05V | 03V05V

Sedangkan pada *Uniform Cross Over* ada penentuan persentase gen yang akan disilangkan misalkan 50%, angka ini nantinya perlu masukkan dari *user*.

Individu 1 = 02H04H | 05V04H 03V

Individu 2 = 01H02H | 01V03V05V

Setelah disilangkan, akan dilakukan pengecekan terhadap masing-masing individu, apakah terjadi pengulangan. Kedua individu yang telah disilangkan ini diperbaiki, sehingga tidak ada pengulangan lagi. Garis besarnya adalah setiap angka yang diulang ditukar dengan pasangannya, yaitu angka yang diulang di kromosom pasangannya.

Hasil untuk *Two Points Cross Over*:

Individu 1 = 01H05H01V04H03V

Individu 2 = 02H04H05V03V05V

Hasil untuk *Uniform Cross Over*:

Individu 1 = 02H01H05V04H 03V

Individu 2 = 04H02H01V03V05V

2.5.7 Tahap Mutasi (*Mutation*)

Cara lain untuk mendapatkan individu yang baru yaitu dengan mutasi. Mutasi pertama yang mungkin terjadi adalah perubahan urutan barang. Hal ini dilakukan secara acak, diambil 2 angka (nomor barang) dari satu individu, kemudian ditukar.

Asal : 06H05H01H04V02V03H

Hasil: 06H02V01H04V05H03H

Pada contoh ini, angka 2 dan 5 (barang kedua dan kelima dari daftar barang) ditukar urutannya.

Mutasi kedua yang mungkin terjadi adalah perubahan posisi barang ketika dimasukkan ke dalam gudang, dari posisi horisontal menjadi posisi vertikal, atau sebaliknya.

Asal : 01H02H04H03H06V05H

Hasil: 01H02H04H03V06V05H

2.5.8 Perulangan Tahap

Satu populasi baru telah terbentuk dengan selesainya mutasi. Populasi baru tersebut akan menjadi populasi awal bagi generasi selanjutnya, dan GA akan mengulang tahap 2 sampai 5 secara terus menerus sampai sejumlah generasi yang telah ditentukan.

```
2 Kode, KodeGudang : String[5];
3 Nama              : String[10];
4 X,Y,Z,X2,Z2       : SmallInt;
5 P,L,T              : Byte;
6 Berat             : Integer;
7 KodeRak, KodeFloor : Byte;
8 TSetBarang = Array[1..80]Of TBarang;
```

Struktur data *floor*:

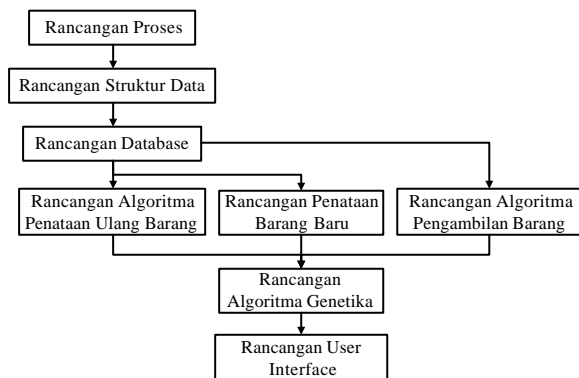
```
1 TLRak = Array[1..20,1..20] Of Byte;
2 Tfloor = Record
3 T, TA : SmallInt;
4 Kondisi : TLRak;
5 Isi : Array[1..50] Of TBarang;
6 JumIsi : Byte;
7 BebanMax,BebanNow : Integer;
```

Struktur data Rak:

```
1 TRak = Record
2 X,Y : SmallInt;
3 P,L : SmallInt;
4 Floor : Array[1..6] Of TFloor;
5 JumFlr : Byte;
6 ARTrak = Array[1..20] Of TRak;
```

3. RANCANGAN SISTEEM

Bab ini akan membahas rancangan sistem dari perangkat lunak yang dibuat, dimana akan ditampilkan sejumlah *flowchart* dari perangkat lunak yang dibuat.



Gambar 6. Rancangan Sistem Perangkat Lunak

3.1 Struktur Data

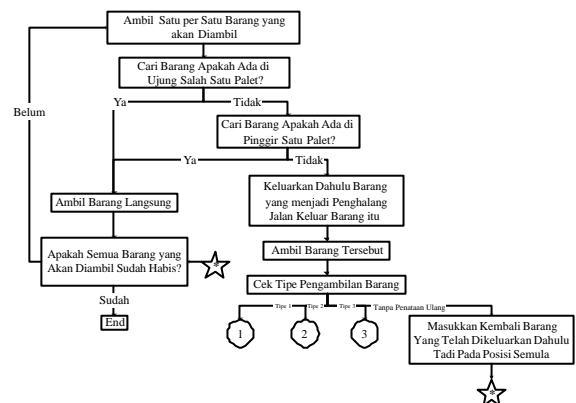
Struktur data gudang:

```
1 TGudang = Record
2 Kode : String; //Kode gudang
3 Panjang:SmallInt; //Panjang gudang
4 Lebar : SmallInt; //Lebar gudang
5 Tinggi : SmallInt; //Tinggi gudang
```

Struktur data barang:

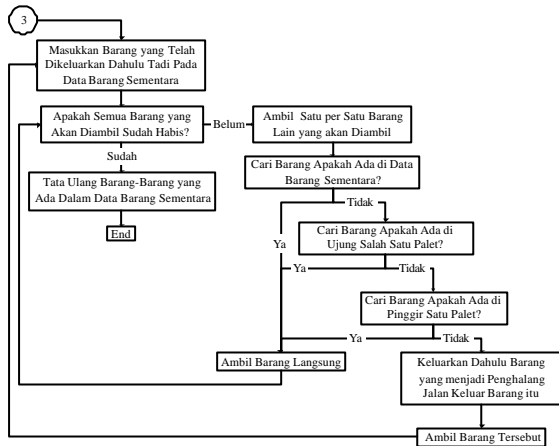
```
1 TBarang = Record
```

3.2 Rancangan Algoritma Pengambilan dan Penataan Ulang Barang



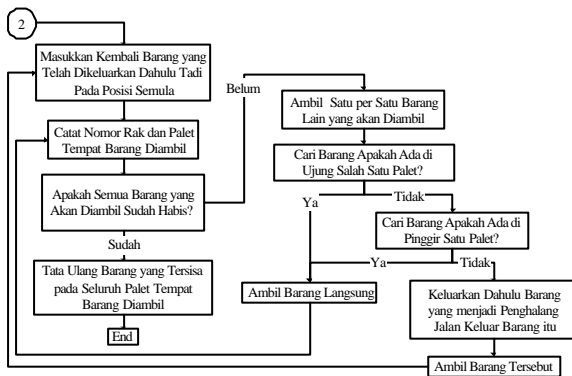
Gambar 7. Algoritma Pengambilan dan Penataan Ulang Barang

3.2.1 Jalan Keluar Barang



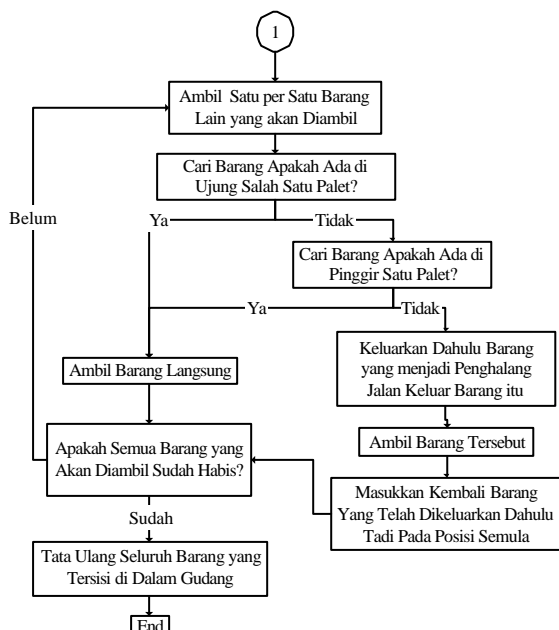
Gambar 8. Algoritma Penataan Barang Keluar

3.2.2 Tata Ulang Satu Palet



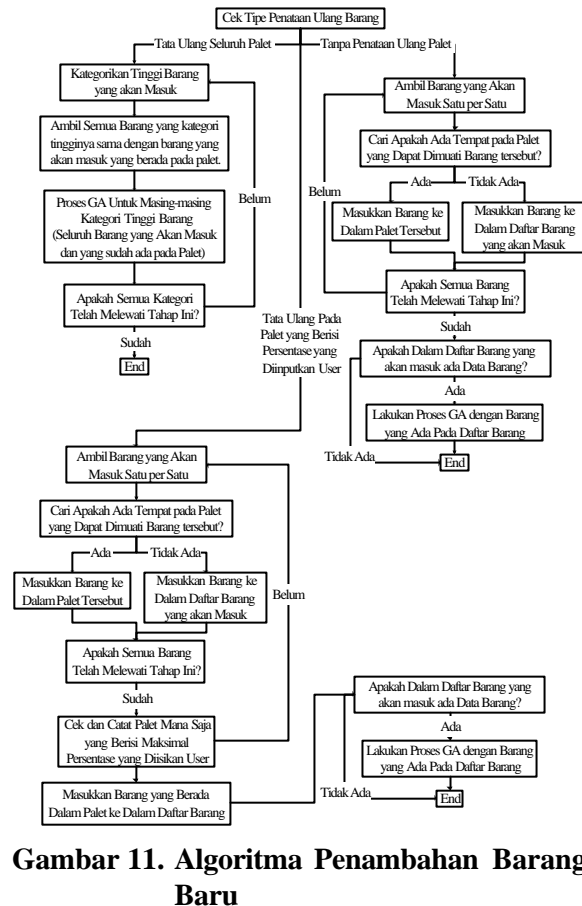
Gambar 9. Algoritma Penataan Ulang satu Palet

3.2.3 Penataan Ulang Seluruh Gudang



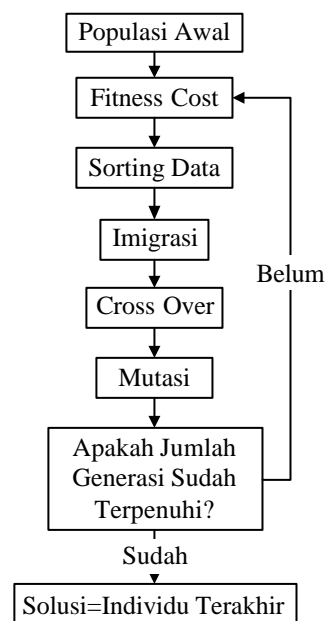
Gambar 10. Algoritma Penataan Ulang Seluruh Gudang

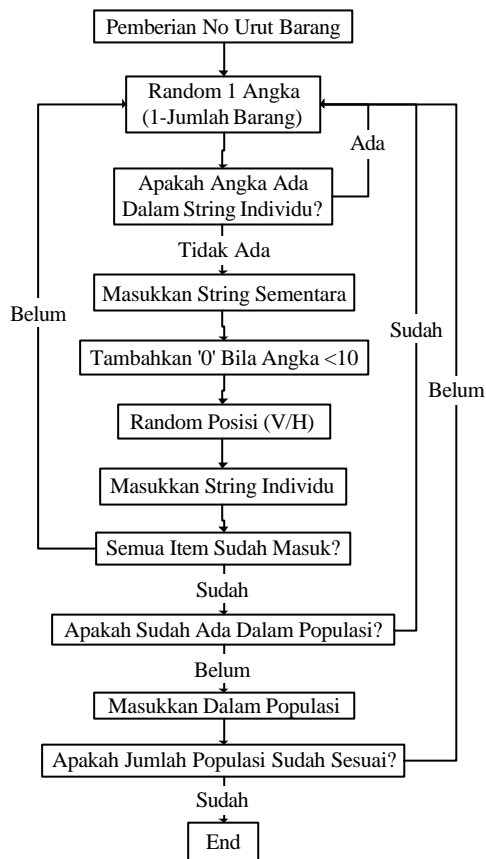
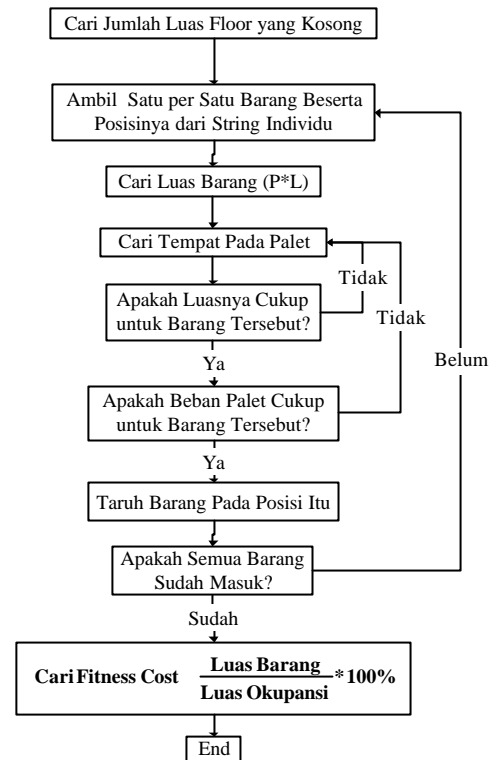
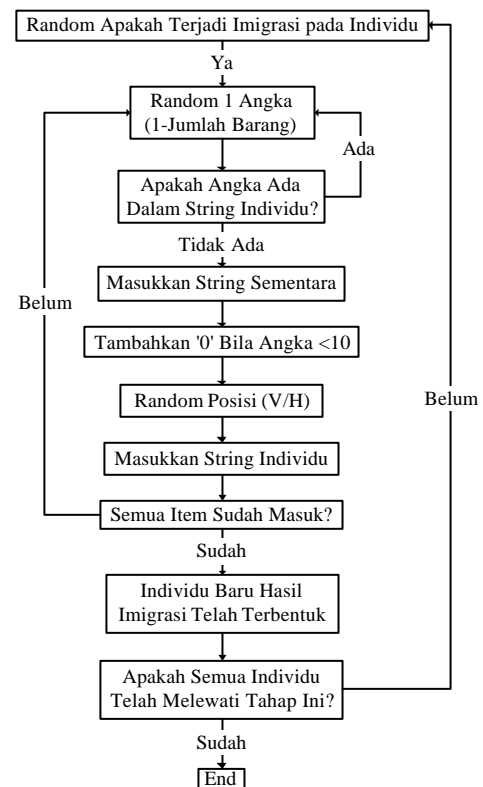
3.3 Rancangan Penambahan Barang Baru

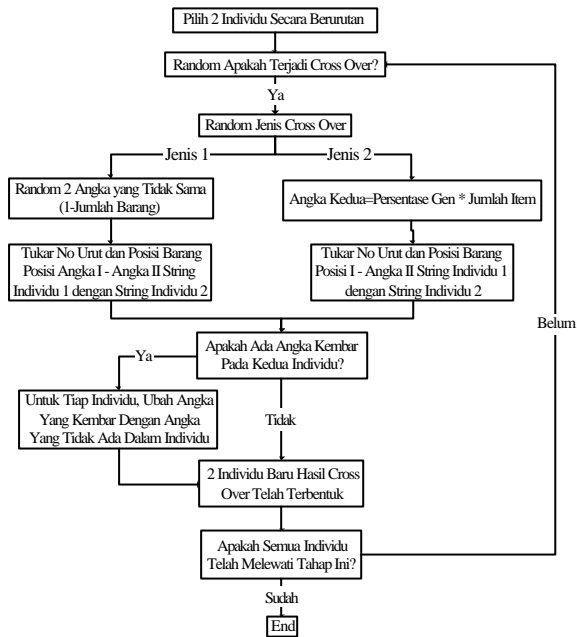


Gambar 11. Algoritma Penambahan Barang Baru

3.4 Rancangan Algoritma Genetika

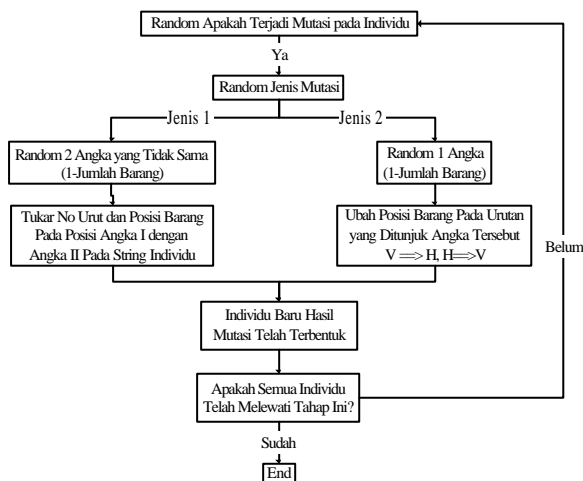


Gambar 12. Algoritma Genetika**3.4.1 Populasi Awal****Gambar 13. Algoritma Pembuatan Populasi Awal****3.4.2 Fitness Cost****Gambar 14. Algoritma Pencarian Fitness Cost****3.4.3 Imigrasi****Gambar 15. Algoritma Imigrasi**
3.4.4 Cross Over



Gambar 16. Algoritma Cross Over

3.4.5 Mutasi



Gambar 17. Algoritma Mutasi

4. PENGUJIAN

Pada bab ini akan dilakukan pengujian penataan barang, pengambilan barang, dan memasukkan barang baru ke dalam gudang.

4.1 Penataan Barang

Berikut barang-barang yang akan ditata dalam gudang:

Tabel 2. Barang 1

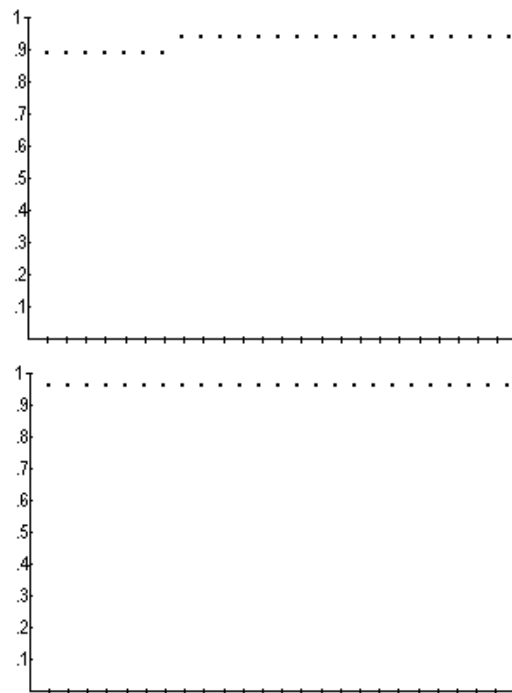
Kode	Panjang	Lebar	Tinggi	Jumlah	Berat
001	20	30	40	8	3
002	50	20	30	5	4
003	20	10	30	4	7
004	40	30	10	2	15
005	50	30	10	2	12
006	20	30	80	7	5
007	40	20	60	4	10
008	60	40	90	9	8

Sementara parameter-parameter GA yang akan dipakai dalam seluruh pengujian ini adalah sebagai berikut:

Tabel 3. Parameter GA

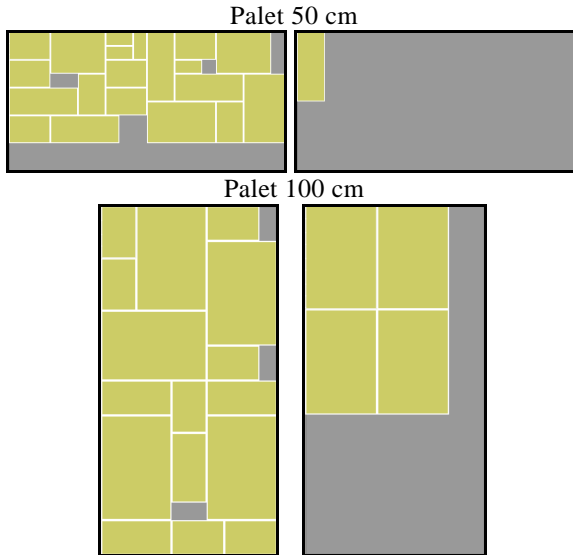
Populasi	Gene-rasi	Immigra-tion Rate	Mutation Rate	Cross Over Rate	Cross Over Gen Rate
25	25	0.5	0.5	1	0.5

Grafik *fitness cost* hasil pengujiannya adalah:

Gambar 18. Grafik *Fitness Cost* 1

Fitness cost dari pengujian ini untuk pelat 50 cm adalah 94% (gambar 18 kiri) dan untuk pelat 100 cm adalah 98% (gambar 18 kanan), Walaupun barang-barang yang dimasukkan ke gudang beraneka ragam jenisnya, perangkat lunak ini tetap bisa menjalankan fungsinya dengan baik.

Kondisi paletnya dapat dilihat pada gambar di bawah ini. Di dalam pengujian ini dan pengujian-pengujian selanjutnya palet 50 cm akan diletakkan secara horisontal dan vertikal untuk palet 100cm.



Gambar 19. Kondisi Palet 1

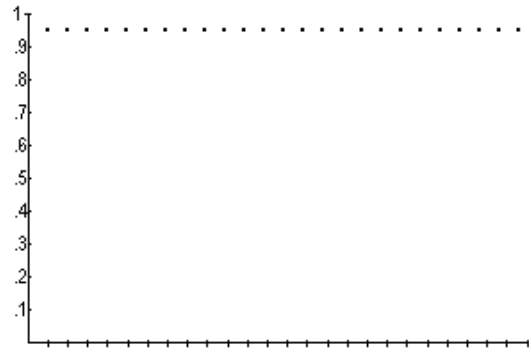
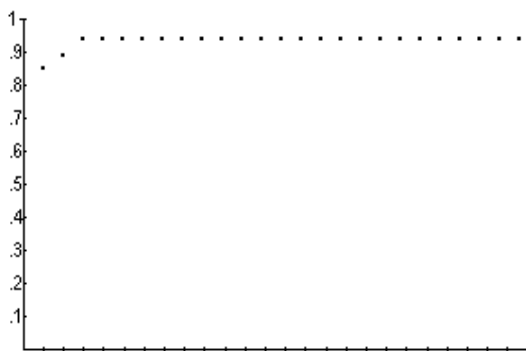
4.2 Pengambilan Barang Penataan ulang Seluruh Gudang

Barang-barang yang akan diambil adalah:

Tabel 4. Barang 2

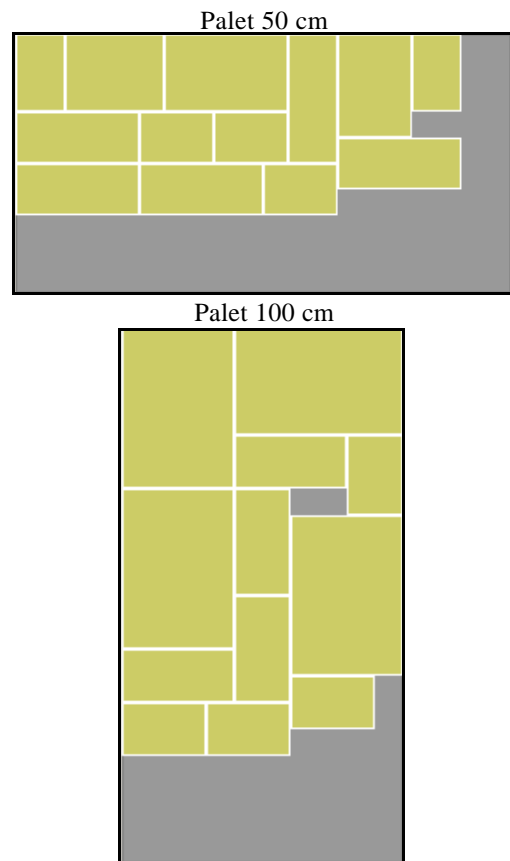
Kode	Jumlah
001	3
003	4
005	1
006	3
008	5

Grafik *fitness cost* dan keadaan palet dari pengujian ini dapat dilihat pada gambar 20 berikut ini:



Gambar 20. Grafik *Fitness Cost* 2

Hasil penataan ulang ini menghasilkan *fitness cost* 94% (Gambar 20 Kiri) dan 95% (Gambar 20 Kanan) untuk palet dengan tinggi 50 cm dan 100 cm. Berikut keadaan pelatnya:



Gambar 21. Kondisi Palet 2

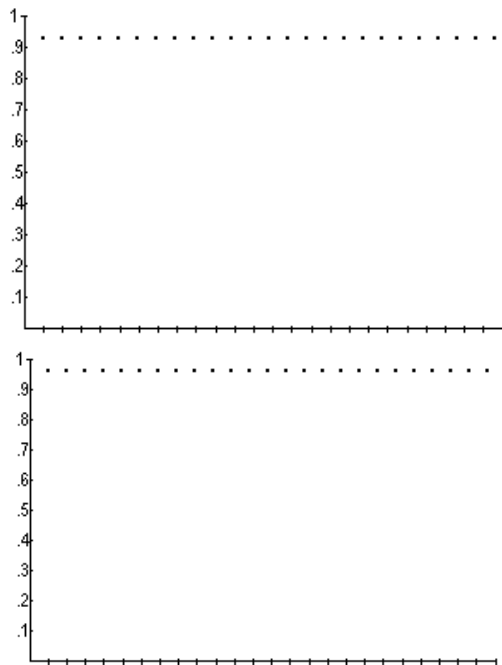
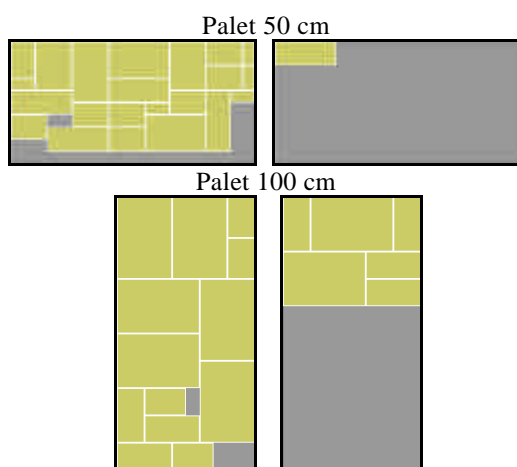
4.3 Memasukkan Barang Baru dengan Penataan Seluruh Palet

Barang yang akan dimasukkan ke dalam gudang adalah sebagai berikut:

Tabel 5. Barang 3

Kode	P	L	T	Jumlah	Berat
001	20	30	40	3	3
003	20	10	30	5	7
005	50	30	10	2	12
007	40	20	60	3	10
008	60	40	90	4	8

Fitness Cost yang didapatkan dari pengujian ini adalah 93% (Gambar 22 Kiri) dan 97% (Gambar 22 Kanan) masing-masing untuk palet dengan tinggi 50 cm dan 100 cm. Berikut grafik beserta keadaan paletnya:

**Gambar 22. Grafik *Fitness Cost* 3****Gambar 23. Kondisi Palet 3**

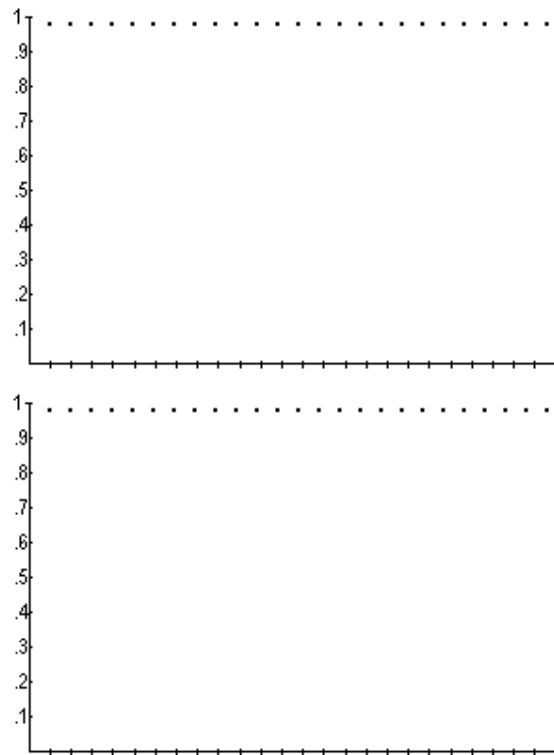
4.4 Pengambilan Barang dengan Penataan Ulang Palet

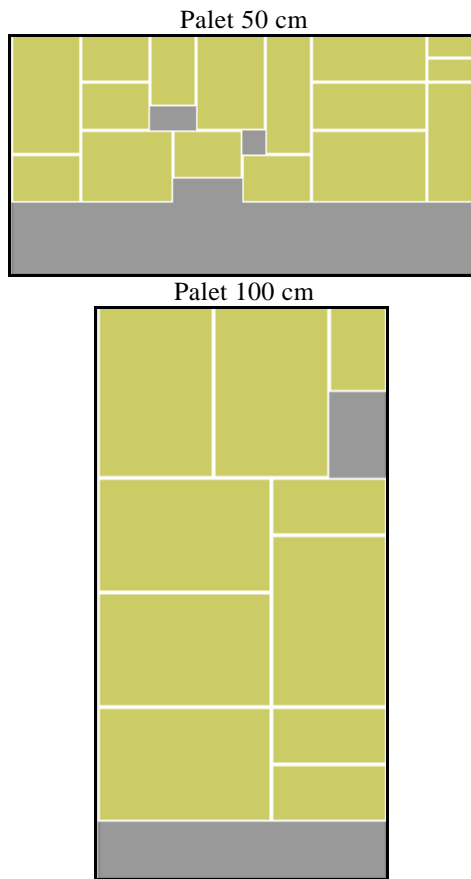
Barang yang akan diambil:

Tabel 6. Barang 4

Kode	Jumlah
001	2
003	3
005	1
006	3

Fitness cost untuk palet 50 cm adalah 97% (Gambar 24 Kiri), dan 98% untuk palet 100 cm (Gambar 24 Kanan). Berikut gambar grafik dan keadaan paletnya:

**Gambar 24. Grafik *Fitness Cost* 4**



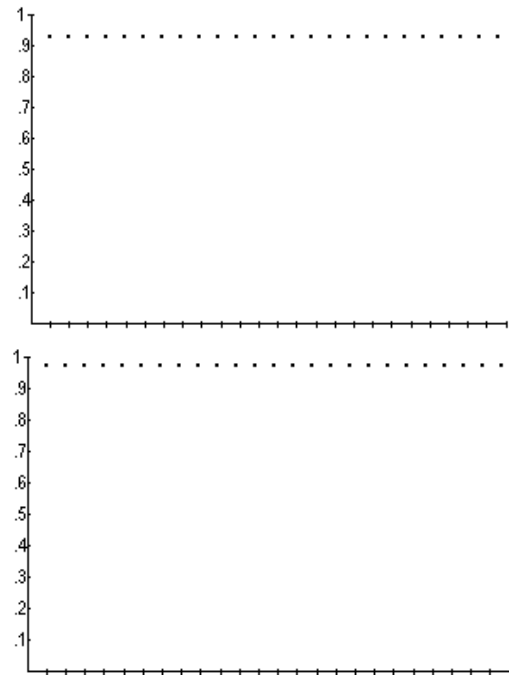
Gambar 25. Kondisi Palet 4

4.5 Memasukkan Barang Baru dengan Penataan Palet < 50%

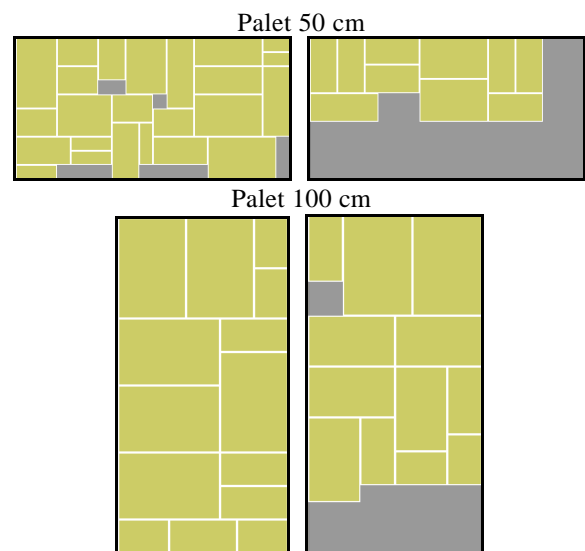
Kali ini barang yang akan dimasukkan adalah sebagai berikut:

Tabel 7. Barang 5

Kode	Jumlah
001	2
003	4
005	3

Gambar 26. Grafik *Fitness Cost 5*

Grafik di atas menunjukkan angka yang cukup maksimal yaitu 93% (Gambar 26 Kiri) untuk palet 50 cm, dan 97% (Gambar 26 Kanan) untuk palet 100 cm. Gambar berikut menunjukkan kondisi paletnya.



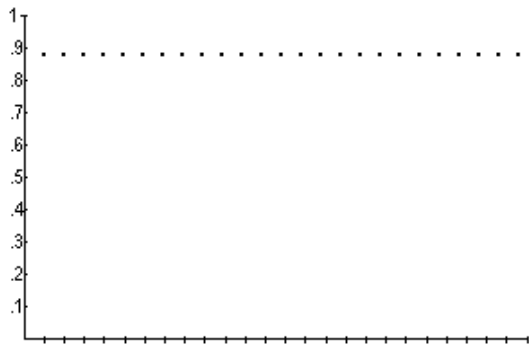
Gambar 27. Kondisi Palet 5

4.6 Pengambilan Barang dengan Penataan Ulang Barang yang Dikeluarkan Dahulu

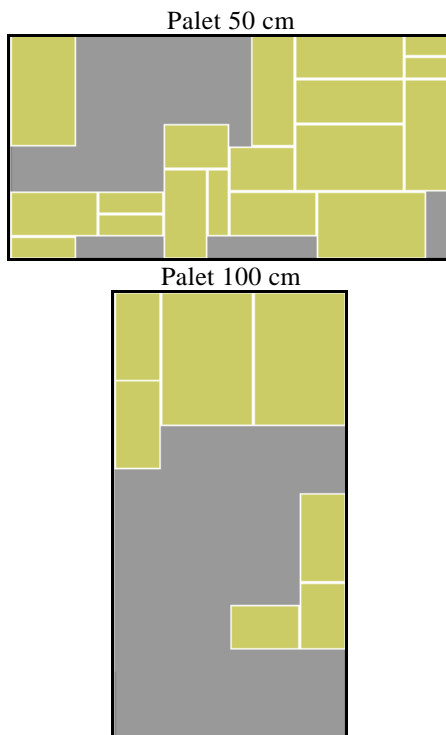
Barang-barang yang akan diambil selanjutnya adalah:

Tabel 8. Barang 6

Kode	Jumlah
001	4
004	2
011	5

**Gambar 28. Grafik Fitness Cost 6**

Karena barang yang perlu dikeluarkan dahulu hanya ada pada palet 100 cm, maka grafik *fitness cost* di atas hanya untuk palet 100 cm saja dan grafik ini menunjukkan angka yang agak rendah yaitu 88%, hal ini tentu saja dikarenakan hanya menata ulang 1 barang saja sementara barang yang lain tetap pada posisinya masing-masing. Berikut keadaan paletnya:

**Gambar 29. Grafik Fitness Cost 6**

5. KESIMPULAN

Kesimpulan-kesimpulan yang didapat dari serangkaian pengujian yang telah dilakukan pada bab sebelumnya antara lain:

- Algoritma genetika merupakan algoritma yang dapat memberikan solusi yang baik dengan waktu yang cukup singkat.
- Penentuan nilai-nilai parameter algoritma genetika sangat mempengaruhi hasil akhir yang akan didapatkan.
- Algoritma genetika akan memiliki solusi yang lebih baik pada jumlah populasi dan jumlah generasi yang tinggi.
- Solusi yang didapat dari algoritma genetika akan selalu berbeda-beda dan tidak menentu, hal ini disebabkan oleh karena besarnya unsur random yang ada dalam algoritma ini.
- Pengambilan barang yang paling efektif dari posisi ujung-ujung palet. Selain mudah mengambilnya, juga tidak perlu dilakukan penataan ulang.
- Penataan ulang barang yang memberikan hasil yang paling optimal adalah penataan ulang seluruh isi gudang.
- Perangkat lunak yang dibuat dapat dikatakan sudah memenuhi kriteria dalam pengisian dan penataan ulang barang ke gudang dengan sistem rak, karena solusi-solusi yang diberikan cukup baik. Kelayakan penggunaan perangkat lunak di atas 90%, hal ini dibuktikan dengan serangkaian pengujian yang telah dilakukan pada bab sebelumnya.

DAFTAR PUSTAKA

1. C. Pimpawat, N. Chaiyaratana, *Using a co-operative co-evolutionary genetic algorithm to solve a three-dimensional container loading problem*. May 2001. PDF
2. Darrell Whitley, A Genetic Algorithm Tutorial, http://samizdat.mines.edu/ga_tutorial/

3. Jeff Molofee's Open GL Tutorials, <http://nehe.gamedev.net/tutorials/>, 1997.
4. Melanie Mitchell, *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*
5. Michael D. Vose, *The Simple Genetic Algorithm: Foundations and Theory (Complex Adaptive Systems)*
6. Genetic Server and Genetic Library, <http://www.nd.com/genetic/>
7. James C. Bean, *A Probabilistic Model of The Multiple-Choice Genetic Algorithm*, University of Michigan, Ann Arbor, 1997
8. Mulcahy, David E., *Warehouse Distribution and Operations Handbook*, McGraw-Hill, Inc, 1994
9. Open GL Tutorial, <http://www.eecs.tulane.edu/www/Terry/OpenGL/>, 1997
10. The Genetic Algorithm, <http://www.pmsi.fr/gainita.htm>
11. Erwin Kristianto Julistiono, *Optimasi Pengisian Gudang dengan Penerapan Stack menggunakan Metode Genetic Algorithm*, 2003